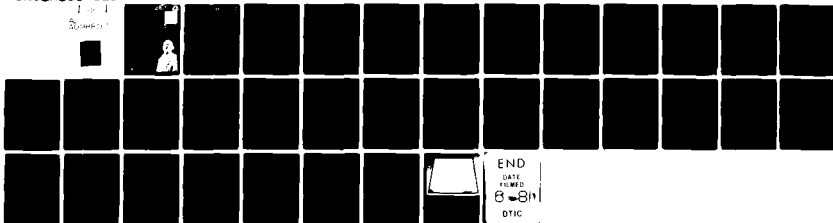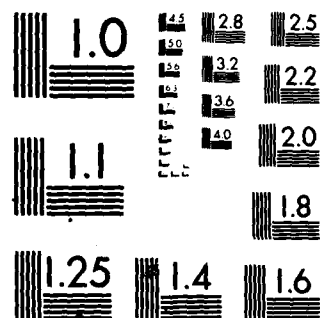AD-A098 807   GEORGE WASHINGTON UNIV WASHINGTON D C PROGRAM IN LOG--ETC F/G 12/1
NUMERICAL METHODS FOR TRANSIENT SOLUTIONS OF MACHINE REPAIR PRO--ETC(U)
JAN 81 H ARSHAM, A R BALANA, D GROSS   N00014-75-C-0729
UNCLASSIFIED   SERIAL-T-436   NL

END
DATE
FILMED
8 –81
DTIC

1.0

1.1

1.25 1.4 1.6

2.8 2.5
3.2 2.2
3.6
4.0 2.0

1.8

4.5
5.0
5.6
6.3

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

THE
GEORGE
WASHINGTON
UNIVERSITY

STUDENTS FACULTY STUDY R
ESEARCH DEVELOPMENT FUT
URE CAREER CREATIVITY CC
MMUNITY LEADERSHIP TECH
NOLOGY FRONTIE SIGN
ENGINEERING APP ENC
GEORGE WASHIN NI\

SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

NUMERICAL METHODS FOR TRANSIENT SOLUTIONS
OF MACHINE REPAIR PROBLEMS

by

Hossein Arsham
Arturo R. Balana
Donald Gross

Serial T-436
5 January 1981

The George Washington University
School of Engineering and Applied Science
Institute for Management Science and Engineering

81 5 12 013

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SERIAL-T-436	2. GOVT ACCESSION NO. AD-A098807	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NUMERICAL METHODS FOR TRANSIENT SOLUTIONS OF MACHINE REPAIR PROBLEMS		5. TYPE OF REPORT & PERIOD COVERED SCIENTIFIC rept.
		6. PERFORMING ORG. REPORT NUMBER T-436
7. AUTHOR(s) HOSSEIN ARSHAM ARTURO R. BALANA DONALD GROSS		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0729
9. PERFORMING ORGANIZATION NAME AND ADDRESS THE GEORGE WASHINGTON UNIVERSITY PROGRAM IN LOGISTICS WASHINGTON, DC 20052		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS OFFICE OF NAVAL RESEARCH CODE 434 ARLINGTON, VA 22217		12. REPORT DATE 5 Jan 1981
		13. NUMBER OF PAGES 29
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) NONE
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC SALE AND RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

FINITE SOURCE QUEUES
MACHINE REPAIR PROBLEMS
TRANSIENT QUEUES

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper reviews and compares three numerical methods of computing transient probabilities of finite Markovian queues (particularly the machine repair problem). A brief review of each method is followed by a numerical example of a moderate size machine repair problem (two-stage cyclic queue).

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73
S/N 0102-014-6601 |

406 337

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Program in logistics

Abstract
of
Serial T-436
5 January 1981

NUMERICAL METHODS FOR TRANSIENT SOLUTIONS
OF MACHINE REPAIR PROBLEMS

by

Hossein Arsham
Arturo R. Balana
Donald Gross

This paper reviews and compares three numerical methods of computing transient probabilities of finite Markovian queues (particularly the machine repair problem). A brief review of each method is followed by a numerical example of a moderate size machine repair problem (two-stage cyclic queue).

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Program in Logistics

NUMERICAL METHODS FOR TRANSIENT SOLUTIONS
OF MACHINE REPAIR PROBLEMS

by

Hossein Arsham
Arturo R. Balana
Donald Gross

1. Introduction

The classic machine repair problem with spares is a typical
example of a finite state space queueing problem and consists of a fixed
number of identical machines of which initially M are operating and Y
are spares. The M machines are in parallel and are independent. When
one fails, it is instantaneously replaced by a spare if a spare is
available; if not, less than M machines will operate until a repaired
machine becomes available. Simultaneously, the failed machine goes
instantaneously into a repair facility.

1.1 Assumptions and Problem Statement

The following assumptions are made concerning the machine repair
example.

(a) The system failure rate is proportional to the number of
operating machines.

(b) Each machine has exponential failure time with mean $1/\lambda$.

(c) There are c parallel servers in the repair facility.

(d) Each server has exponential service time with mean $1/\mu$.

Thus, the machine repair problem is Markovian and the states of this Markov process can be described by a single number i , where i represents the number of machines in the repair station. The intensity matrix of this process is given by

$$Q \equiv \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & \cdots & & 0 \\ \mu_1 & -\lambda_1-\mu_1 & \lambda_1 & 0 & & 0 \\ 0 & \mu_2 & -\lambda_2-\mu_2 & \lambda_2 & & 0 \\ \vdots & & & & & \vdots \\ 0 & & & & \mu_{M+Y} & -\mu_{M+Y} \end{bmatrix}$$

where

$$\lambda_i = \begin{cases} M\lambda & , & (0 \le i < Y) \\ (M-i+Y)\lambda & , & (Y \le i \le Y+M) \\ 0 & , & (i \ge Y+M) \end{cases}$$

$$\mu_i = \begin{cases} i\mu & , & (0 \le i \le c) \\ c\mu & , & (i \ge c) \end{cases} .$$

For this problem, steady state solutions in closed form are readily attainable [see, for example, Gross and Harris (1974)].

1.2 Transient Solutions

It is desired to find the transient solutions for the machine repair problem. If the problem has N states (N = M+Y+1) , the intensity matrix provides N equations

$$\Pi'(t) = \Pi(t) \cdot Q , \tag{1}$$

where $\Pi(t)$ is a N = M+Y+1 component vector whose elements are $\pi_i(t)$, the unconditional probability that the system is in state i at

- 2 -

time t , and $\Pi'(t)$ a vector of derivatives of $\pi_i(t)$. Finding solutions in closed form [see, for example, Marlow (1978)] is extremely difficult and in most cases impossible (unless M+Y is very small).
However, a variety of procedures is available which can yield numerical solutions to the differential equations. In this paper we will discuss several of these numerical procedures and attempt to apply each procedure to a machine repair problem with the following parameters:

$M=4$, number of machines initially operating

$Y=1$, number of spares

$c=2$, number of service channels

$\lambda=.15$, machine failure rate

$\mu=.5$, service rate.

Under the assumptions mentioned in the preceding section one can obtain the following initialized first order system of differential equations. Assuming at t=0 all machines are up:

$$
\begin{bmatrix} \pi_0'(t) \\ \pi_1'(t) \\ \pi_2'(t) \\ \pi_3'(t) \\ \pi_4'(t) \\ \pi_5'(t) \end{bmatrix} = \begin{bmatrix} \pi_0(t) \\ \pi_1(t) \\ \pi_2(t) \\ \pi_3(t) \\ \pi_4(t) \\ \pi_5(t) \end{bmatrix}^T \begin{bmatrix} -.6 & .6 & 0 & 0 & 0 & 0 \\ .5 & -1.1 & .6 & 0 & 0 & 0 \\ 0 & 1 & -1.45 & .45 & 0 & 0 \\ 0 & 0 & 1 & -1.3 & .3 & 0 \\ 0 & 0 & 0 & 1 & -1.15 & .15 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}
$$

with initial value $\Pi(0) = [1,0,0,0,0,0]$.

2. Randomization Techniques

The randomization procedures give a method of calculating the transition probability matrix $P(t)$, i.e., the order (M+Y+1) square matrix whose elements are $p_{ij}(t)$, the probability that the system is in state j at time t given that it started in state i $(0 \leq i,j \leq M+Y)$. From $P(t)$, by using the initial probability vector $\Pi(0)$,

$\Pi(t)$ can be calculated from $\Pi(t) = \Pi(0)P(t)$. We denote the $(i,j)th$ element of Q by q_{ij}, and define a scalar β as

$$\beta = \max_i |q_{ii}| . \tag{2}$$

Now define a matrix P with elements p_{ij} as

$$P = \{p_{ij}\} \equiv I + \frac{1}{\beta} Q ,$$

where I is an identity matrix having the same order as Q.

The matrix P is stochastic, and it has been shown [see Cohen (1969)] that the elements of $P(t)$ can be obtained by

$$p_{ij}(t) = e^{-\beta t} \sum_{n=0}^{\infty} \frac{\beta^n t^n}{n!} p_{ij}^{(n)} , \tag{3}$$

where $p_{ij}^{(n)}$ is the $(i,j)th$ element of matrix P raised to the nth power. This method is called randomization because it can be inter-preted as a discrete time Markov chain with transition probabilities p_{ij} and transition time generated by a Poisson process at rate β.

To compute $p_{ij}(t)$ involves raising the matrix P to the nth power for $n=0,1,2,\ldots$. The numerical procedure truncates n at some appropriate value, say m, so

$$p_{ij}(t) = e^{-\beta t} \sum_{n=0}^{m-1} \frac{\beta^n t^n}{n!} p_{ij}^{(n)} + R_m , \tag{4}$$

where R_m is the error due to truncation.

2.1 Barzily and Gross Method

Barzily and Gross (1979) proposed a criterion to truncate n such that for a given $\epsilon > 0$, the smallest m is chosen such that the error R_m obeys

- 4 -

$$R_m \leq 1 - e^{-\beta t} \sum_{n=0}^{m-1} \frac{(\beta t)^n}{n!} \leq \varepsilon . \qquad (5)$$

Their paper shows that such an m exists. For a machine repair problem
with parameters $M = Y = C = 1$ and $\varepsilon = 10^{-3}$, the result of applying
this criterion has been shown to be quite satisfactory, in fact accurate
up to the second decimal place. A complete description of the Barzily-
Gross method together with the study of the transient effects and the
speed of convergence to steady state for machine repair problems can be
found in the above cited reference. The algorithm has been coded in
FORTRAN to run on the IBM 3031 computer, under the program name WONG.
The following section is a brief description of program WONG.

2.2 Program WONG

WONG, a FORTRAN code originally designed to find the spares in-
ventory level and number of repair channels necessary to guarantee a
prespecified service level for a machine repair problem, included in it
a program to provide transient solutions for the system state probabil-
ities. This portion of the program was separated from the original and
updated to stand alone as a provider of transient solutions to machine
repair problems. Input requirements for program WONG are given in
Appendix 1. Results of applying program WONG to the sample problem of
Section 1.2 are given in Section 5.

3. The QUE Package

Grassmann and Servranckx (1979) developed a FORTRAN based package
for finding transient solutions for moderate sized queueing networks (up
to ten state variables). The method adopted in this package is in fact
based on the randomization procedure discussed in the preceding sections.
The truncation criteria are somewhat different and are described in the
following section. We have adopted the sample problem of Section 1.2 to
the specifications of this package and the results are presented, along
with those of program WONG, in Section 5.

Grassmann (1977) has also shown that the truncation error, R_m , can be bounded with reasonable accuracy; that is,

$$R_m \leq 1 - \sum_{n=0}^{m-1} \frac{(\beta t)^n e^{-\beta t}}{n!} . \tag{6}$$

For small βt , the sum in the RHS of (6) can easily be evaluated for fixed m , and thus m can be determined such that R_m will be below a prescribed value $\varepsilon > 0$, as was done by Barzily and Gross (1979). For large βt one can approximate the Poisson distribution by the normal distribution. In the QUE package, m is set equal to

$$m = \beta t + 4\sqrt{\beta t} + 5 . \tag{7}$$

Using Poisson tables for $\beta t \leq 20$, or normal tables otherwise this procedure guarantees that such an m yields an R_m less than 10^{-4} .

3.1 Formulation of the Sample Problem for the QUE Package

To solve the machine repair problem by using the QUE package, one must formulate the problem to fit the network structure input requirements. The output statistics are then obtained from the program. One way of formulating the problem to fit the QUE package requirements is shown in Figure 1.
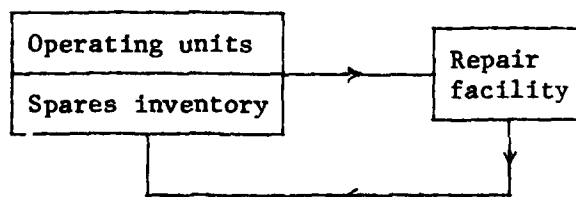

Figure 1.--Formulation of the sample machine repair problem for the QUE package.

It is necessary to define state variables and event descriptions together with type, conditions, net effect, and rate parameters. These are shown in Table 1.

TABLE 1

THE EVENT DESCRIPTION OF THE PROBLEM FOR THE QUE PROGRAM

Events	Type	Rate	Condition	Net Effect
1. Arrival into repair station				
a. when no machine is in repair station	1	.6	$X1=0$	(+1)
b. when there are some machines in the repair station	1	$.75 - .15X1$	$1 \leq X1 \leq 4$	(+1)
2. Service: when only one repairman is busy	1	.5	$X1=1$	(-1)
3. Service: when both repairmen are busy	1	1	$2 \leq X1 \leq 5$	(-1)

The state variable is described by $X1 \equiv$ number of machines in repair station, $0 \leq X1 \leq 5$.

Each event has a rate function which associates the rate of each transition with the starting state. The rate function may be constant or a function of the state variable. Types of events are classified as type one, having finite rate event, or as type two, having infinite rate event. The state space of the system is defined by general conditions represented by linear inequalities involving the state variables. The net effect is the value of the state variable after the event occurs and is determined by incrementing or decrementing the value by a constant prior to the event's occurrence. A more detailed explanation can be found in Grassmann and Servanckx (1979). Appendix 2 shows the QUE program input requirements for the sample problem.

4. Numerical Integration Methods

Numerical integration methods can be employed to solve a system of ordinary differential equations described by

$$Y'(t) \equiv \begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_p'(t) \end{bmatrix} = \begin{bmatrix} f_1(y_1,\ldots,y_p;t) \\ f_2(y_1,\ldots,y_p;t) \\ \vdots \\ f_p(y_1,\ldots,y_p;t) \end{bmatrix} \equiv f(Y,t) \qquad (8)$$

with known initial value $Y(t_0)$. The standard techniques are generally variations of either Runge-Kutta (R-K) or predictor-corrector (p-c) methods.

Runge-Kutta methods are based on formulas that approximate the Taylor series solutions

$$y_i(t+h) = y_i(t) + h \cdot y_i'(t) + \frac{h^2}{2} y_i''(t) + \ldots + \frac{h^k}{k!} y_i^{(k)}(t) + R_k ,$$

$i=1,\ldots p$. These methods use approximations for the second and higher-order derivatives. Euler's method is a special R-K method, with $k=1$. These methods have been used by several authors [e.g., Bookbinder and Martell (1979), Grissmann (1977), Liitschwager and Ames (1975) and Neuts (1975)] to find transient solutions in queueing systems.

The predictor-corrector methods require information about several previous points in order to evaluate the next point. These methods involve using one formula to predict the next $Y(t)$ value, followed by the application of a more accurate corrector formula. Unlike the R-K methods, p-c methods are not self-starting; hence, they use the R-K method to obtain the first $Y(t)$ value.

Predictor-corrector methods can provide an estimate of the local truncation error at each step in the calculations, in contrast to the R-K methods, which cannot obtain such an estimate.

Predictor-corrector methods include Milne's method, Hamming's method, and Adams' methods. Methods based on the Adams formulas have performed very well in test problems [see Hull, Enright, Fellen, and Sedgwick (1972)] even for nonstiff[1] systems or when function evaluations are relatively expensive. Hull *et al.* also concluded that R-K methods are not competitive, although fourth or fifth order methods are best for problems in which function evaluations are not very expensive and accuracy requirements are not very stringent.

Predictor-corrector methods have been used by Ashour and Jha (1973) for queueing problems.

A variety of routines is available for solving a system of ordinary differential equations. They include RKGS, DRKGS (fourth order R-K formulas), HPCG, DHPCG, HPCL, DHPCL (Hamming's Methods), all from the IBM Scientific Subroutine Package [IBM (1968)]; and DVERK (Verner's fifth and sixth order R-K formulas) and DVOGER (Gear's Method) in the International Mathematical and Statistical Libraries (IMSL) package [IMSL (Ed. 6)]. One routine based on extrapolation methods is DREBS, also in the IMSL package, which uses the Bulvisch-Stoer method.

4.1 Gear's Algorithm

C. W. Gear (1971a, 1971b) proposed a variable-order integration method based on Adams' predictor-corrector formulas of orders one through seven. It uses an order one formula to start and, for this reason, must start with very small step size when the error tolerance is stringent.

Gear's algorithm includes a special approach for dealing with stiff differential equations.

---

[1] A stiff system of ordinary differential equations is characterized by the property that the ratio of the largest to the smallest eigenvalue is much greater than one.

The Adams' formulas fall under two general categories—open and closed formulas. The Adams-Bashforth pth order formulas (open formula) can be written as

$$y_n = y_{n-1} + h \sum_{K=1}^{P} \beta_K y'_{n-K} , \qquad (9)$$

where $y_i = y(t_i)$, $t_i = ih$, $y'_i = f(y_i, t_i)$. The order of the method is one less than the order of the truncation error per step. The Adams-Moulton pth order formulas (closed formulas) can be written as

$$y_{n,m} = y_{n-1} + \beta_0^* h f\left(y_{n,m-1}, t_n\right) + h \sum_{K=1}^{p-1} \beta_K^* y'_{n-K} . \qquad (10)$$

The coefficients β_K and β_K^* are given by Henrici (1962). Equation (9) is used as the first approximation in Equation (10). Thus (9) is used as the predictor equation and (10) as the corrector equation. Whenever (10) converges (as is true when h is small and f is smooth), the truncation error introduced at the nth integration step is $c_{p+1}^A h^{p+1} y^{(p+1)}(t_n) + O(h^{p+2})$, where $y^{(K)}$ is the kth derivative of y , and c_{p+1}^A are constants [see Henrici (1962)].

The predictor equation (9) is equivalent to fitting a pth degree polynomial through the known quantitites y_{n-1}, hy'_{n-1}, $\ldots$, hy'_{n-p} . For more details of the algorithm, see Gear (1971a, 1971b).

4.2 DVOGER Subroutine

DVOGER is a FORTRAN routine based on Gear's algorithm designed to solve a set of first order ordinary differential equations. The algorithm chooses the order of approximation such that the step size is increased, thereby decreasing the solution time. The option of using a particular method is done through a switch variable (MTH). Results of using DVOGER on the sample problem are also given in section 5. Appendix 3 shows the input and programming requirements for exercising DVOGER.

5. <u>Conclusions and Numerical Results</u>

We have presented three numerical methods in computing the transient probabilities for finite Markovian queues. Transient probabilities often provide a realistic picture of actual queueing systems, and sometimes it is desirable to know how fast they converge to steady state [Barzily and Gross (1979)].

The methods considered in this paper fall into two categories, namely, randomization and predictor-corrector numerical integration. In general these methods give reasonably accurate results for a moderate sized problem. Table 2 shows the output of these programs for $t=1,3,5,7,9,12$. The QUE package and DVOGER show almost equal results, while program WONG deviates from the other two by at most 3×10^{-4}, which is reasonably compatible.

In terms of set-up effort, program WONG gives the least degree of difficulty since it was written primarily for machine repair problem. The biggest concern with respect to the QUE program was the huge core storage requirement, which exceeds the current daytime capacity of 384K bytes of the IBM 3031 at the GWU Center for Academic and Administrative Computing. Future modification by redimensioning is suggested. In using DVOGER, one must carefully choose applicable parameter values, as in the step size. Total running times of the programs are 2.35 seconds for program QUE, 6.13 seconds for program WONG, and 162.62 seconds for DVOGER. The reason for the length of the latter is that with step size fixed at 1×10^{-4}, DVOGER must be called 120,000 times to integrate for each time point from $t=0$ to $t=12$. There is a need to explore further the best options of DVOGER to find those which might reduce running time considerably.

TABLE 2

PROGRAM OUTPUT OF WONG, QUE, DVOGER FOR THE TRANSIENT SOLUTION
OF A MACHINE REPAIR PROBLEM WITH PARAMETERS M=4, Y=1, C=2,
λ=.15, μ=.5 and $\Pi(0)$=(1,0,0,0,0,0) AT t=1,3,5,7,9,12

Time	Program	$\pi_0(t)$	$\pi_1(t)$	$\pi_2(t)$	$\pi_3(t)$	$\pi_4(t)$	$\pi_5(t)$
1	WONG	.6235	.2946	.0708	.0096	.0007	.0000
	QUE	.6237	.2949	.0709	.0097	.0007	.0000
	DVOGER	.6237	.2949	.0709	.0097	.0007	.0000
3	WONG	.3944	.3686	.1722	.0536	.0099	.0008
	QUE	.3945	.3688	.1723	.0536	.0099	.0008
	DVOGER	.3945	.3688	.1723	.0536	.0099	.0008
5	WONG	.3331	.3663	.2005	.0781	.0192	.0022
	QUE	.3333	.3665	.2006	.0782	.0192	.0022
	DVOGER	.3333	.3665	.2006	.0782	.0192	.0022
7	WONG	.3119	.3618	.2091	.0886	.0244	.0033
	QUE	.3122	.3620	.2093	.0887	.0244	.0033
	DVOGER	.3122	.3621	.2093	.0887	.0244	.0033
9	WONG	.3038	.3595	.2123	.0931	.0269	.0038
	QUE	.3039	.3597	.2124	.0932	.0269	.0038
	DVOGER	.3039	.3597	.2124	.0932	.0269	.0038
12	WONG	.2995	.3580	.2138	.0955	.0283	.0042
	QUE	.2997	.3582	.2140	.0956	.0284	.0042
	DVOGER	.2997	.3582	.2140	.0956	.0284	.0042

6. Acknowledgments

APPENDIX 1: PROGRAM WONG

A. Input Requirements

TABLE A1.1

PARAMETER CARD INPUT

Columns	Format	Input Name	Explanation
1-5	I5	M	Number of machines initially operating
6-10	I5	IC	Number of service channels
11-15	I5	IY	Number of spares
16-27	F12.7	RLAM	Machine failure rate (Poisson mean)
28-39	F12.7	RMU	Service rate (Poisson mean)
40-49	F10.6	EPS	Tolerance value

TABLE A1.2

TIME INPUT

Columns	Format	Input Name	Explanation*
1-7	F7.3	TDEL	Time T

*Time at which transient probability is
required is format free, but it must be coded
starting from column one, and a separate card
is required for every time desired.

B. Numerical Example Input

We shall illustrate the use of program WONG on the sample problem
given in Section 1.2. The cards for the sample problem, with $\epsilon = 10^{-3}$

- 13 -

and t = 3,5 are shown in Figure A1.1. The output obtained for this problem is tabulated in Table A1.3.

Card Type	Card Image
Job	// STANDARD JOB CARD
JCL	// EXEC FØRT2
JCL	//FØRT.SYSIN DD *
Program Deck	[Program WONG deck]
JCL	//GØ.SYSIN DD *
Parameter	4 2 1 .15 .5 .001
Time	3
Time	5
JCL	//

Figure A1.1—Card input program WONG for sample problem.

TABLE A1.3

THE OUTPUT OF PROGRAM WONG
FOR THE SAMPLE PROBLEM*

t	$\pi(t)$					
	$\pi_0'(t)$	$\pi_1(t)$	$\pi_2(t)$	$\pi_3(t)$	$\pi_4(t)$	$\pi_5(t)$
3	.3944	.3686	.1722	.0536	.0099	.0008
5	.3331	.3663	.2005	.0781	.0192	.0022

*Note: the initial distribution of the system is assumed to be $\pi_0(0)=1$, $\pi_i(0)=0$, $i \geq 1$.

APPENDIX 2:   PROGRAM QUE

A. Input Deck Format Specification

TABLE A2.1

INPUT DECK FORMAT FOR PROGRAM QUE

1. Problem title card:
    Function: for documentation only

Columns	Format	Field description
1-80	20A4	Problem title

2. Problem specification card:
    Function: describes the number of state variables, number of
    events, and the number of general system conditions

Columns	Format	Field description
1-2	I2	Number of state variables
3-4	I2	Number of events
5-6	I2	Number of state space restrictions

3. Maximum vector card (one card for each state variable—for
    machine repair problems only one is required)
    Function: to describe the highest possible value of each state
    variable (a maximum of ten state variables)

Columns	Format	Field description
1-2	I2	Maximum value of state variable 1

4. Event title cards (one for each event)
    Function: for documentation purposes only

Columns	Format	Field description
1-80	20A4	Event title

TABLE A2.1--*continued*

5. Event specification card (one for each event)
   Function: indicates type of event and the rate of this event

Columns	Format	Field description
1	I1	Print flag for transitions (1 = Yes)
2	I1	Event type (1 for machine repair problem)
3	I1	Number of specific conditions (0 for machine repair problem)
4	I1	Flag for new minima and maxima of the state variable (1 = Yes)
5-10	F6.2	Rate of this event
11-12	I2	State variable on which rate depends (if zero, rate is a constant)
13-18	F6.2	Increase of rate

6. New maxima and minima vector card (one for each event)
   Function: resets the maximum and minimum values for the state variable

Columns	Format	Field description
1-2	I2	New maximum for state variable X1
3-4	I2	New minimum for state variable X1

7. Net effect card
   Function: defines the function f(x) which converts the starting state into the target state

Columns	Format	Field description
1-3	I3	Net effect for state variable X1

8. Trailer card
   Function: delimiter card used to indicate the end of events section of the system's input. The first four bytes of the record must contain the string "END"

Columns	Format	Field description
1-4	A4	Control field (value is "END")
5-80	19A4	Ignored

TABLE A2.1--*continued*

9. Probability specification card
   Function: gives the number of nonzero initial probabilities
   (only the nonzero ones need to be entered)

Columns	Format	Field description
1-2	I2	Number of nonzero initial probabilities
3-8	F6.2	The starting time of the system

10. Initial probability card
    Function: specified initial probability and the state to which
    it pertains (one for each state variable--only one required for
    machine repair problem)

Columns	Format	Field description
1-6	F6.5	Initial probability
7-8	I2	State variable X1

11. Time specification card
    Function: to indicate the time for which transient solutions are
    required, to indicate what measures are to be printed, and to
    give a criterion whether or not to continue calculating the
    times on the following card

Columns	Format	Field description
1	I1	Number of times for which solutions are required (5 or less)
2	I1	Print flag; if value is 1, joint distributions are printed
3	I1	Print flag; if value is 1, marginal distributions are printed
4	I1	Print flag; if value is 1, expectations are printed
5	I1	If value is 1, other cards follow (having the same format as this one)
6-11	F6.5	Stopping criterion (accuracy desired)

12. Time card
    Function: gives each time (a maximum of five) for which results
    are desired

TABLE A2.1--*continued*

Columns	Format	Field description
1-5	F5.2	First time t_1
6-10	F5.2	Second time t_2
⋮	⋮	⋮
the times $t_1, t_2, \ldots$		must be input in increasing order of magnitude

B. Numerical Example Input

Input data for the sample machine repair problem is shown in Table A2.2.

TABLE A2.2

INPUT CARDS FOR PROGRAM QUE

Card Type	Card Input
1	TRANSIENT SOLUTION FOR MACHINE REPAIR PROBLEM
2	010400
3	05
4	ARRIVAL INTO REPAIR STN: (A) NO MACHINE
5	1101000.60
6	C000
7	001
4	ARRIVAL INTO REPAIR STN: (B) 1 MACHINE
5	1101000.7501-00.15
6	0401
7	001
4	SERVICE WHEN ONE REPAIRMAN IS IDLE
5	1101000.50
6	0;01
7	-01

TABLE A2.2--*continued*

Card Type	Card Input
4	SERVICE WHEN BOTH REPAIRMEN ARE BUSY
5	1101001.00
6	0502
7	-01
8	END OF EVENT SECTION
9	01000.00
10	1.000000
11	41111.0001
12	00.2500.5000.7501.00
11	41111.0001
12	01.2501.5001.7502.00
11	41111.0001
12	02.2502.5002.7503.00
11	41111.0001
12	03.2503.5003.7504.00
11	41111.0001
12	04.2504.5004.7505.00
11	41111.0001
12	05.2505.5005.7506.00
11	41111.0001
12	06.2506.5006.7507.00
11	41111.0001
12	07.2507.5007.7508.00
11	41111.0001
12	08.2508.5008.7509.00
11	41111.0001
12	09.2509.5009.7510.00
11	41111.0001
12	10.2510.5010.7511.00
11	41111.0001
12	11.2511.5011.7512.00
11	00000

APPENDIX 3: DVOGER SUBROUTINE

A. Input and Options

This section discusses the input requirements for DVOGER. Since DVOGER is a library subroutine, one must write a computer program in order to use it. The advantage is in the flexibility of inputting the parameter values, as well as in the choice of output variables, frequency of printing the solutions, and so forth.

The input structure is as follows.

1. Job and JCL cards. See Section B for the standard and job control language cards.

2. Main program. The main or the calling program is to be written in FORTRAN. The proper dimensioning of arrays, the input mode of parameter values, the number of calls to DVOGER, and the frequency of printing the solution must be determined by the user. Moreover, an external subroutine DFUN is to be written by the user to compute functional values $F(y,t)$ or the Jacobian of $F(y,t)$.

The parameters needed for the main program include:

N = number of first order differential equations

M = order of Jacobian $(M=N)$

T = initial value of independent variable (e.g., time)

MTH = method indicator

$= \begin{cases} 0, \text{ predictor-corrector (Adams) method} \\ 1, \text{ variable-order method, suitable for stiff} \\ \quad \text{systems (partial derivatives provided by user)} \\ 2, \text{ variable-order method (partial derivatives com-} \\ \quad \text{puted by numerical differencing)} \end{cases}$

- 20 -

Y(1,N) = an input array of initial solutions at T ; array Y is to be declared 8×N

YMAX(N) = an input array of maximum absolute value of solution

HMIN = smallest step size allowable

HMAX = maximum step size allowable

H = step size to be attempted on the next step; this is to be used if it does not cause a larger error than requested

JSTART = $\begin{cases} -1, \text{ repeat the last step with a new } H \\ 0, \text{ initialize the integration (for first call to DVOGER)} \\ 1, \text{ take a new step continuing from the last} \end{cases}$

EPS = maximum error criterion such that the single step error estimates divided by YMAX(I) are less than EPS in norm.

The call to DVOGER is done by the statement,

CALL DVOGER (DFUN, Y, T, N, MTH, MAXDER, JSTART, H, HMIN, HMAX, EPS, YMAX, ERROR, WK, IER).

3. Subroutine DFUN. DFUN is user-supplied and is to be declared by an EXTERNAL DFUN statement in the main or calling program. DFUN specifies the problem for DVOGER. It provides the system of equations and the Jacobian. The parameters include:

YP(1,N) = vector of solution TP

TP = present time

M = order of Jacobian

IND = $\begin{cases} 0, \text{ DFUN computes } F(YP,TP) \\ 1, \text{ DFUN computes Jacobian of } F \text{ evaluated at } (YP,TP) \end{cases}$

YP is to be declared as an 8×N array.

B. Numerical Example Input

We shall illustrate the solution of the sample problem using the DVOGER subroutine. The structure of a single job run is as follows.

(a) Job and JCL cards

    // STANDARD JOB CARD

    //   EXEC FØRG2

    //FØRT.SYSIN DD *

    [Main Program and Subroutine]

    //GØ.SYSLIB   DD

    //   DD DSN=GWU.IMSL6.LMØD.D,DISP=SHR

    //   DD DSN=GWU.IMSL6.LMØD.S,DISP=SHR

    //GØ.SYSIN DD *

(b) Figure A3.1 shows the main program and subroutine DFUN, and the
     input and our options, where the transient solutions are re-
     quired for times T=1,2, and 3 .

(c) Program Output. The step size h was fixed at 0.0001 by
     specifying HMAX=HMIN=H=0.0001 . The method used was the
     predictor-corrector method based on the Adams formulas
     (MTH=0) . The output of transient solutions was printed out
     at every time increment of .01 starting at t=0 to t=3 .
     The error tolerance was set at 10^{-7} .

The program below was written for this specific problem, although
a more general program can be written to handle any problem with arbi-
trary parameter values.

The authors have not tested this subroutine for options which give
minimum execution time, as this was not the purpose here.

```
      DIMENSION YP(8,10),PW(10,10),Y(8,10),YMAX(10),DY(10)
      DIMENSION WK(140,20),ERROR(10),C(10)
      DOUBLE PRECISION YP
      DOUBLE PRECISION C,D,E,H,P,T,Y,R1,BND,EPS,EUP,EDWN,ENQ1,D1,D2,
     *                 ENQ2,ENQ3,HMAX,HMIN,HNEW,HOLD,TOLD,YMAX,
     *                 ERROR,RACUM,WK,XK,ZERO,HALF,ONE,ONEP,PW
      EXTERNAL DFUN
      KK=0
      N=6
      M=6
      T=0.0D0
      Y(1,6)=0.0D0
      Y(1,5)=0.0D0
      Y(1,4)=0.0D0
      Y(1,3)=0.0D0
      Y(1,2)=0.0D0
      Y(1,1)=1.0D0
      YMAX(1)=1.0D0
      YMAX(2)=1.0E0
      YMAX(3)=1.0D0
      YMAX(4)=1.0D0
      YMAX(5)=1.0D0
      YMAX(6)=1.0D0
      JSTART=0
      IND=0
      MTH=0
      HMAX=1.0D-4
      HMIN=1.0D-4
      H=1.0D-4
      EPS=1.0D-7
      WRITE(6,100)
100   FORMAT('0',9X,'T',12X,'P0',12X,'P1',12X,'P2',12X,'P3',
     * 12X,'P4',12X,'P5')
      WRITE(6,200) T,Y(1,1),Y(1,2),Y(1,3),Y(1,4),Y(1,5),Y(1,6)
      DO 10 K=1,30000
      CALL DVOGER(DFUN,Y,T,N,MTH,MAXDER,JSTART,H,HMIN,HMAX,EPS,
     * YMAX,ERROR,WK,IER)
      KK=KK+1
      IF(KK.NE.100) GOTO 250
      WRITE(6,200) T,Y(1,1),Y(1,2),Y(1,3),Y(1,4),Y(1,5),Y(1,6)
200   FORMAT('0',7(2X,F10.8))
      KK=0
250   CONTINUE
      H=1.0D-2
      IND=0
      MTH=0
      IF(MTH.NE.1) GOTO 10
      PW(1,1)=-.6
      PW(1,2)=.5
      PW(1,3)=0.0
```

**Figure A3.1--Program listing to call DVOGER for the machine repair
problem,   t=1,2 and 3 .**

```
            PW(1,4)=0.0
            PW(1,5)=0.0
            PW(1,6)=0.0
            PW(2,1)=.6
            PW(2,2)=-1.1
            PW(2,3)=1.0
            PW(2,4)=0.0
            PW(2,5)=0.0
            PW(2,6)=0.0
            PW(3,1)=0.0
            PW(3,2)=.6
            PW(3,3)=-1.45
            PW(3,4)=1.0
            PW(3,5)=0.0
            PW(3,6)=0.0
            PW(4,1)=0.0
            PW(4,2)=0.0
            PW(4,3)=.45
            PW(4,4)=-1.3
            PW(4,5)=1.0
            PW(4,6)=0.0
            PW(5,1)=0.0
            PW(5,2)=0.0
            PW(5,3)=0.0
            PW(5,4)=.3
            PW(5,5)=-1.15
            PW(5,6)=1.0
            PW(6,1)=0.0
            PW(6,2)=0.0
            PW(6,3)=0.0
            PW(6,4)=0.0
            PW(6,5)=.15
            PW(6,6)=-1.0
         10 CONTINUE
            STOP
            END
            SUBROUTINE DFUN(YP,TP,M,DY,PW,IND)
            DIMENSION PW(10,10),YP(8,10),DY(10)
            DOUBLE PRECISION YP,TP,DY,PW
            IF(IND.EQ.0) GOTO 5
            PW(1,1)=-.6
            PW(1,2)=.5
            PW(1,3)=0.0
            PW(1,4)=0.0
            PW(1,5)=0.0
            PW(1,6)=0.0
            PW(2,1)=.6
            PW(2,2)=-1.1
            PW(2,3)=1.0
            PW(2,4)=0.0
            PW(2,5)=0.0
            PW(2,6)=0.0
            PW(3,1)=0.0
            PW(3,2)=.6
            PW(3,3)=-1.45
```

Figure A3.1--*continued*

```
      PW(3,4)=1.0
      PW(3,5)=0.0
      PW(3,6)=0.0
      PW(4,1)=0.0
      PW(4,2)=0.0
      PW(4,3)=.45
      PW(4,4)=-1.3
      PW(4,5)=1.0
      PW(4,6)=0.0
      PW(5,1)=0.0
      PW(5,2)=0.0
      PW(5,3)=0.0
      PW(5,4)=0.3
      PW(5,5)=-1.15
      PW(5,6)=1.0
      PW(6,1)=0.0
      PW(6,2)=0.0
      PW(6,3)=0.0
      PW(6,4)=0.0
      PW(6,5)=.15
      PW(6,6)=-1.0
      GOTO 10
    5 CONTINUE
      DY(1)=-.6*YP(1,1)+.5*YP(1,2)
      DY(2)=.6*YP(1,1)-1.1*YP(1,2)+YP(1,3)
      DY(3)=.6*YP(1,2)-1.45*YP(1,3)+YP(1,4)
      DY(4)=.45*YP(1,3)-1.3*YP(1,4)+YP(1,5)
      DY(5)=.3*YP(1,4)-1.15*YP(1,5)+YP(1,6)
      DY(6)=.15*YP(1,5)-YP(1,6)
   10 RETURN
      END
```

Figure A3.1--*continued*

REFERENCES

ASHOUR, S. and R. D. JHA (1973). Numerical transient-state solutions of queueing systems. *Simulation*, 21, 117-122.

BARZILY, Z. and D. GROSS (1979). Transient solutions for repairable item provisioning. Technical Paper Serial T-390, Program in Logistics, The George Washington University.

BOOKBINDER, J. H. and P. L. MARTELL (1979). Time-dependent queueing approach to helicopter allocation for forest fire initial attack. *INFOR*, 58-70.

COHEN, J. W. (1969). *The Single Server Queue*. North-Holland, London and Wiley, New York.

GEAR, C. W. (1969). The automatic integration of stiff ordinary differential equations. *Information Processing 68.* (A.J.H. Morrell, ed.). North-Holland, Amsterdam, 187-193.

GEAR, C. W. (1971a). DIFSUB for solution of ordinary differential equations. *Comm. ACM*, 14, 185-190.

GEAR, C. W. (1971b). The automatic integration of ordinary differential equations. *Comm. ACM*, 14, 176-179.

GRASSMANN, W. (1977). Transient solutions in Markovian queueing system. *Computers and Operations Res.*, 4, 47-56.

GRASSMANN, W. and J. SERVRANCKX (1979). The QUE package. University of Saskatchewan.

GROSS, D. and C. M. HARRIS (1974). *Fundamentals of Queueing Theory*. Wiley, New York.

HENRICI, P. (1962). *Discrete Variable Methods in Ordinary Differential Equations*. Wiley, New York.

HULL, T.; W. ENRIGHT; B. FELLEN; and A. SEDGWICK (1972). Comparing numerical methods for ordinary differential equations. *SIAM Journal on Numerical Analysis*, 9, (4), 603-637.

IBM CORPORATION (1968). System/360 Scientific Subroutine Package Version III. White Plains, New York, IBM Corporation.

IMSL. International Mathematical and Statistical Library, IBM Edition 6.

LIITSCHWAGER, J. and W.F. AMES (1975). On transient queues--practice and pedagogy. *Proc. 8th Symp. Interface*, Los Angeles, California, 206.

MARLOW, W. H. (1978). *Mathematics for Operations Research*. Wiley, New York.

NEUTS, M. F. (1975). Algorithms for the waiting time distribution under various queue disciplines in the M/G/1 queue with service time distribution of phase type. Department of Statistics, Mimeograph Series, No. 420.

THE GEORGE WASHINGTON UNIVERSITY

Program in Logistics

Distribution List for Technical Papers

The George Washington University
  Office of Sponsored Research
  Gelman Library
  Vice President H. F. Bright
  Dean Harold Liebowitz
  Dean Henry Solomon

ONR
  Chief of Naval Research
    (Codes 200, 434)
  Resident Representative

OPNAV
  OP-40
  DCNO, Logistics
  Navy Dept Library
  NAVDATA Automation Cmd

Naval Aviation Integrated Log Support

NARDAC Tech Library

Naval Electronics Lab Library

Naval Facilities Eng Cmd Tech Library

Naval Ordnance Station
  Louisville, Ky.
  Indian Head, Md.

Naval Ordnance Sys Cmd Library

Naval Research Branch Office
  Boston
  Chicago
  New York
  Pasadena
  San Francisco

Naval Ship Eng Center
  Philadelphia, Pa.

Naval Ship Res & Dev Center

Naval Sea Systems Command
  PMS 30611
  Tech Library
  Code 073

Naval Supply Systems Command
  Library
  Operations and Inventory Analysis

Naval War College Library
  Newport

BUPERS Tech Library

FMSO

USN Ammo Depot Earle

USN Postgrad School Monterey
  Library
  Dr Jack R. Borsting
  Prof C. R. Jones

US Coast Guard Academy
  Capt Jimmie D. Woods

US Marine Corps
  Commandant
  Deputy Chief of Staff, R&D

Marine Corps School Quantico
  Landing Force Dev Ctr
  Logistics Officer

Armed Forces Industrial College

Armed Forces Staff College

Army War College Library
  Carlisle Barracks

Army Cmd & Gen Staff College

Army Logistics Mgt Center
  Fort Lee

Commanding Officer, USALDSRA
  New Cumberland Army Depot

Army Inventory Res Ofc
  Philadelphia

Army Trans Material Cmd
  TCMAC-ASDT

Air Force Headquarters
  AFADS-3
  LEXY
  SAF/ALG

Griffiss Air Force Base
  Reliability Analysis Center

Gunter Air Force Base
  AFLMC/XR

Maxwell Air Force Base Library

Wright-Patterson Air Force Base
  AFLC/OA
  Research Sch Log
  AFALD/XR

Defense Technical Info Center

National Academy of Sciences
  Maritime Transportation Res Bd Lib

National Bureau of Standards
  Dr B. H. Colvin
  Dr Joan Rosenblatt

National Science Foundation

National Security Agency

Weapons Systems Evaluation Group

British Navy Staff

National Defense Hdqtrs, Ottawa
  Logistics, OR Analysis Estab

American Power Jet Co
  George Chernowitz

General Dynamics, Pomona

General Research Corp
  Library

Logistics Management Institute
  Dr Murray A. Geisler

Rand Corporation
  Library
  Mr William P. Hutzler

Carnegie-Mellon University
  Dean H. A. Simon
  Prof G. Thompson

Case Western Reserve University
  Prof B. V. Dean
  Prof M. Mesarovic

Cornell University
  Prof R. E. Bechhofer
  Prof R. W. Conway
  Prof Andrew Schultz, Jr.

Cowles Foundation for Research
  in Economics
  Prof Martin Shubik

Florida State University
  Prof R. A. Bradley

Harvard University
  Prof W. G. Cochran
  Prof Arthur Schleifer, Jr.

Princeton University
  Prof A. W. Tucker
  Prof J. W. Tukey
  Prof Geoffrey S. Watson

Purdue University
  Prof S. S. Gupta
  Prof H. Rubin
  Prof Andrew Whinston

Stanford University
  Prof T. W. Anderson
  Prof Kenneth Arrow
  Prof G. B. Dantzig
  Prof F. S. Hillier
  Prof D. L. Iglehart
  Prof Samuel Karlin
  Prof G. J. Lieberman
  Prof Herbert Solomon
  Prof A. F. Veinott, Jr.

University of California, Berkeley
  Prof R. E. Barlow
  Prof D. Gale
  Prof Jack Kiefer

University of California, Los Angeles
  Prof R. R. O'Neill

University of North Carolina
  Prof W. L. Smith
  Prof M. R. Leadbetter

University of Pennsylvania
  Prof Russell Ackoff

University of Texas
  Institute for Computing Science
  and Computer Applications

Yale University
  Prof F. J. Anscombe
  Prof H. Scarf

Prof Z. W. Birnbaum
  University of Washington

Prof B. H. Bissinger
  The Pennsylvania State University

Prof Seth Bonder
  University of Michigan

Prof G. E. Box
  University of Wisconsin

Dr Jerome Bracken
  Institute for Defense Analyses

Continued

February 1981

THE GEORGE WASHINGTON UNIVERSITY
BENEATH THIS PLAQUE
IS BURIED
A VAULT FOR THE FUTURE
IN THE YEAR 2056

THE STORY OF ENGINEERING IN THIS YEAR OF THE PLACING OF THE VAULT AND
ENGINEERING HOPES FOR THE TOMORROWS, AS WRITTEN IN THE RECORDS OF THE
FOLLOWING GOVERNMENTAL AND PROFESSIONAL ENGINEERING ORGANIZATIONS AND
THOSE OF THIS GEORGE WASHINGTON UNIVERSITY

BOARD OF COMMISSIONERS DISTRICT OF COLUMBIA
UNITED STATES ATOMIC ENERGY COMMISSION
DEPARTMENT OF THE ARMY UNITED STATES OF AMERICA
DEPARTMENT OF THE NAVY UNITED STATES OF AMERICA
DEPARTMENT OF THE AIR FORCE UNITED STATES OF AMERICA
NATIONAL ADVISORY COMMITTEE FOR AERONAUTICS
NATIONAL BUREAU OF STANDARDS U S DEPARTMENT OF COMMERCE
AMERICAN SOCIETY OF CIVIL ENGINEERS
AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
THE SOCIETY OF AMERICAN MILITARY ENGINEERS
AMERICAN INSTITUTE OF MINING & METALLURGICAL ENGINEERS
DISTRICT OF COLUMBIA SOCIETY OF PROFESSIONAL ENGINEERS INC
THE INSTITUTE OF RADIO ENGINEERS INC
THE CHEMICAL ENGINEERS CLUB OF WASHINGTON
WASHINGTON SOCIETY OF ENGINEERS
FAULKNER KINGSBURY & STENHOUSE ARCHITECTS
CHARLES H TOMPKINS COMPANY BUILDERS
SOCIETY OF WOMEN ENGINEERS
NATIONAL ACADEMY OF SCIENCES NATIONAL RESEARCH COUNCIL

THE PURPOSE OF THIS VAULT IS INSPIRED BY AND IS DEDICATED TO
CHARLES HOOK TOMPKINS
BECAUSE OF HIS ENGINEERING CONTRIBUTIONS TO THIS UNIVERSITY TO HIS
COMMUNITY TO HIS NATION AND TO THEIR PROGRESS

BY THE GEORGE WASHINGTON UNIVERSITY

ROBERT V FLEMING

To cope with the expanding technology, our society must
be assured of a continuing supply of rigorously trained
and educated engineers. The School of Engineering and
Applied Science is completely committed to this ob-
jective.